NASA-TM-108115

IN-63-TM
135325
p. 12

An Application of Artificial Intelligence to Automatic Telescopes

KEITH SWANSON
MARK DRUMMOND
JOHN BRESINA
ARTIFICIAL INTELLIGENCE RESEARCH BRANCH
MS 269-2
NASA AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035-1000

(NASA-TM-108115) AN APPLICATION OF ARTIFICIAL INTELLIGENCE TO AUTOMATIC TELESCOPES (NASA) 12 p

N93-15291

Unclas

G3/63 0135325

NASA Ames Research Center
Artificial Intelligence Research Branch

Technical Report FIA-92-26

July, 1992

An Application of Artificial Intelligence to Automatic Telescopes

Keith Swanson
swanson@ptolemy.arc.nasa.gov
Tel: 415-604-6016
NASA Ames Research Center
MS: 269-2, Moffett Field, CA 94035-1000, U.S.A.

Mark Drummond *and* **John Bresina**
{drummond,bresina}@ptolemy.arc.nasa.gov
Sterling Software
NASA Ames Research Center
MS: 269-2, Moffett Field, CA 94035-1000, U.S.A.

Abstract

Automatic Photoelectric Telescopes (APTs) allow an astronomer to be removed from the telescope site in both time and space. APTs "execute" an observation program (a set of observation requests) expressed in an ASCII-based language (ATIS) and collect observation results expressed in this same language. The observation program is currently constructed by a Principal Astronomer from the requests of multiple users; the execution is currently controlled by a simple heuristic dispatch scheduler. Research aimed at improving the use of APTs is being carried out by the Entropy Reduction Engine (ERE) project at NASA Ames. The overall goal of the ERE project is the study and construction of systems that integrate planning, scheduling, and control. This paper discusses the application of some ERE technical results to the improvement of both the scheduling and the operation of APTs.

This paper subsumes previous versions which appeared in

1. *Proceedings of the Remote Observing Workshop*, held in Tuscon, AZ in April 1992.
2. *Proceedings of the Steerable Automatic Lunar Ultraviolet Telescope Explorer (SALUTE) Workshop*, held at NASA Ames Research Center, Moffett Field, CA in May 1992.

1 Introduction

Making observations through telescopes is an activity of central importance to NASA. Telescopes have always been a scare resource, and astronomers have had to make do with limited access. Further, an astronomer has been expected to be physically present at a telescope in order to gather data. Restricted access and local operation have limited the amount of data that can be gathered and, thus, have directly contributed to fewer scientific results than might otherwise be expected.

More recently, increasingly sophisticated network and communication technologies have enabled a number of approaches where astronomers may participate in an observation program from a remote location. These approaches range from remote verbal communications with on–site telescope operations staff, to remote "joysticking" of the telescope with real time video feedback. Remote observations provide additional flexibility by allowing the observer to be physically distant, but still in "direct" control of the telescope. While, under this approach, it is no longer necessary for an astronomer to be physically present at the telescope, he or she must still be directly involved during the execution of the observing program.

A further extension of the remote observation protocol allows an astronomer to be removed from the telescope both temporally as well as spatially. While a large space–based telescope is usually operated in this mode (for instance, Hubble), such a mode of operation is now a commercial reality on modest–aperture ground–based telescopes. For example, Fairborn Observatory and AutoScope Corporation have designed and built control systems and associated hardware for the management and control of modest–aperture photoelectric telescopes which can be operated in a fully automated mode. (For a review of these Automatic Photoelectric Telescopes, or APTs, see Genet and Hayes, 1989). While existing automation deals primarily with photoelectric telescopes, support for other sorts of science (*e.g.*, spectroscopy) is currently being studied.

It is clear that not all observation programs can (or even should) be conducted remotely in both time and space, but a surprising number of observation programs would be candidates with the appropriate telescope and automation technologies. While existing automation does not address all needs of all astronomers, it does provide an excellent starting point. In this context, the eventual goal of our project is what we call a "simplified management structure". The term refers to an approach to the management and control of telescopes that minimizes the number of people that must come between an astronomer's scientific goals and the telescopes required to realize those goals. A simplified management structure requires significantly more sophisticated telescope automation than is currently in common use.

The Entropy Reduction Engine (ERE) project, carried out at the Ames Research Center, has been focusing on the construction of integrated planning and scheduling systems. Specifically, the project is studying the problem of integrating planning and scheduling to produce desired behavior specifications (*i.e.*, plans/schedules) that can be continually used (in a closed–loop sense) to guide a system's behavior. The results of this research are particularly relevant when there is some element of dynamism in the environment and, thus, some chance that a previously formed plan will not execute as predicted. We have found that several of the project's technical results appear directly relevant to telescope automation.

This paper reviews some of our technical results, describes a specific telescope automation problem, that of scheduling observations for APTs, and provides a current and forward look at our efforts to improve the scheduling process. The paper is organized as follows. In the next section, we give a brief overview of how APTs are currently scheduled and operated. Following

this, in section 3, we give an ERE project precis, couched primarily in terms of project objectives. Section 4 provides a description of the current status of the system we have constructed to improve APT scheduling and operations, and finally, section 5 outlines where we plan to go with this work.

2 APT problem summary

An Automatic Photoelectric Telescope (APT) is a telescope controlled by a dedicated computer for the purpose of gathering photometric data about various objects in the sky. While there are many sorts of photometric techniques, we focus on the technique known as *aperture photometry*. (An excellent overview of aperture photometry is given by Hall and Genet, 1988). In aperture photometry, a *group* is the primitive unit to be scheduled and executed. A group is a sequence of telescope and photometer commands defined by an astronomer. Any given astronomer has certain scientific goals, and he or she uses the group as the primary unit of instruction to an APT in order to achieve those goals. The language used to define groups is called ATIS (for Automatic Telescope Instruction Set); ATIS is an ASCII-based language for communicating with APTs (the *de facto* standard).

The communication process between astronomer and APT proceeds roughly as follows. First, an astronomer who wishes to use an APT forms a set of groups consistent with his or her scientific goals. Since each telescope can vary (instruments, optical characteristics, mechanical characteristics, location on the Earth), groups must be formulated in terms of a specific telescope. For any given APT, there is a single person who acts as a central clearing-house for usage requests; such a person is known as the APT's *Principal Astronomer*, or PA. Thus, once an astronomer has assembled his or her set of ATIS groups, they send the groups to the appropriate PA. The PA collects together such sets from a variety of astronomers, attempts to ensure the total set of groups is desirable (that the telescope is loaded properly, that user loading is fair, *etc.*), and then sends the complete set of groups off to the computer controlling the telescope. Actual communication between PA and APT is carried out by using personal computers, modems, and phone lines, but the particular technology is not critical for the current discussion. The important aspect of the communication is that the PA can be located anywhere on the planet (in principle) and need only have access to an appropriate communication link.

The PA sends a set of groups to an APT with the intention that these groups should be run for some time; eventually, the PA requests from the telescope the results that have been obtained via the execution of the groups. The elapsed time varies depending on the telescope, the groups, the PA, the using astronomers, and a variety of other factors. The goal is to worry the astronomers (and the PA) as little as possible about the picayune details of day-to-day telescope management. Thus, the telescope is often left alone for significant periods of time (weeks, perhaps months). However long the telescope operates unattended, it is eventually asked for data, and this is returned to the PA as a "results file". The results are also specified in the ATIS language and contain a record of the groups that were executed, relevant observing parameters to help with data reduction, and the raw data obtained from the observations. The PA edits the results file and sends each astronomer the pieces corresponding to the requested observations.

Of course, the interesting part of this process is the part that we've ignored so far; that is, the process by which the groups are selected and executed by the local telescope controller. *This* is the interesting part, and it is with respect to this process that our planning and scheduling work can make a real difference. Currently, a program called ATIScope manages the execution of a file of

groups. ATIScope runs locally at the given telescope, using observatory and telescope sensors to determine when to execute the provided groups. ATIScope has a variety of responsibilities, but we focus specifically on only one of these; namely, *group selection*.

Group selection is accomplished by a test that attempts to find a "currently" executable group. Roughly, a group is executable if the logical preconditions established by its astronomer–creator are met. Typically, these preconditions relate to the current date, current time, and whether the moon is up or down. Additionally, an astronomer can specify a group *priority*, which is used by ATIScope to sort the groups in order of importance. There are other pseudo–preconditions that have to do with frequency of group execution, but we can safely ignore these for now.[1]

Roughly, the core of ATIScope is a sense–check–execute loop. In sensing, all relevant environmental parameters are determined (date, time, moon status). ATIScope next checks to see which of the groups are enabled according to the match between the current sensor values and the astronomer–provided preconditions. We call the set of groups that pass this matching test the *enabled* groups. The set of enabled groups is winnowed by the application of *group selection rules*. These rules express heuristic knowledge relating to the wisdom of executing any particular group before any other. In scheduling parlance, this scheme is sometimes called *heuristic dispatch*, since at any point in time, some task (here, a group) is "dispatched" for execution, and the selection of a task is determined, purely locally, by the application of some domain–specific heuristics. The information content of the heuristics used by ATIScope is not critical for the current discussion (for details, see Genet & Hayes, 1989, pp. 207–210).

In the current context, heuristic dispatch is used to reduce the set of enabled groups into (hopefully) a single group to be executed. If the heuristic group selection rules fail to winnow the set of enabled groups down to a single candidate, then arbitrarily, the first group of the remaining candidates is selected (this, however, almost never happens, as the group selection rules normally produce a single preferred group). Following selection, the lucky group is executed, at which point telescope control is performed per the detailed commands of the astronomer who wrote the group. Of course, there are safety checks to ensure that the astronomer's commands do not damage equipment, but if the commands are well–behaved (and if the weather cooperates), group execution finishes normally, and ATIScope is free to perform another iteration of its sense–check–execute loop.

How well does ATIScope do, in terms of schedule quality, by using this heuristic dispatch technique? There is no question that ATIScope does provide an acceptable level of performance for some astronomers. However, it is clear that telescope performance can be dramatically improved by better group scheduling. With the heuristic dispatch technique, all decisions are *local* in the sense that no temporal look–ahead is performed to evaluate the ramifications of executing a given group. The system also has no memory of what it has done on previous nights, so groups cannot be selected with respect to some desired frequency of execution. Other scheduling techniques, such as those based on *temporal projection* (Drummond, 1989), consider the impact of a given action by looking ahead in time to see how the current local choice impacts global objectives. Look–ahead is only sensible when astronomer objectives can be precisely formulated and relevant dynamics of the execution environment can be reasonably predicted. Assuming that this can be done, it seems clear that a look–ahead scheduler can outperform the current ATIScope heuristic dispatch method. ATIScope, however, provides us with an existing level of performance against which all would–be contenders can be gauged.

[1] The main factors that influence frequency of execution are a group's *probability* and *number of observations*; see Genet & Hayes (1989), p. 208.

3 Planning, Scheduling, and Control

The design of systems that can synthesize plans has been a long standing research topic in the field of Artificial Intelligence (AI). Such systems, called *planners*, are given a description of the problem at hand and synthesize a plan to solve that problem. Of course, a plan is merely a specification of a solution, and so must be executed to *actually solve* the given problem. Various sorts of "execution system" are possible; for instance, a plan might be executed by a manufacturing system, by a group of people, or by a robotic device; all that is required is a system that is capable of realizing the plan's actions and producing the desired result. The design of these automatic planners has been addressed in AI since its earliest days, and a large number of techniques have been introduced in progressively more ambitious systems over the years. As part of the AI research branch at NASA Ames, the Entropy Reduction Engine (ERE) project is our focus for extending these classical techniques in a variety of ways. In this section, we present the ERE project's overall goals; for more detail on the architecture itself, see Bresina & Drummond (1990), and Drummond, Bresina, & Kedar (1991).

The Entropy Reduction Engine project addresses research on planning and scheduling in the context of closed–loop plan execution. The eventual goal of the ERE project is a set of software tools for designing and deploying integrated planning and scheduling systems that are able to effectively control their environments. Our overall project has two important theoretical subgoals: first, we are working to integrate *planning* and *scheduling*; second, we are studying plan execution as a problem of *discrete event control*. The following paragraphs consider these complementary goals.

The first subgoal is a theoretical and practical integration of planning and scheduling. Traditional AI planning deals with the selection of actions that are relevant to achieving given goals. Various disciplines, principally Operations Research, and more recently AI, have been concerned with the scheduling of actions; that is, with sequencing actions in terms of metric time and metric resource constraints. Unfortunately, most of the work in scheduling remains theoretically and practically disconnected from planning. Consider: a scheduling system is given a set of actions and returns, if possible, a schedule composed of those actions in some specific order. If the scheduler cannot find a satisfactory schedule, then it simply fails. The business of planning is to *select* actions that can solve a given problem, so what we need is an integrated planning and scheduling system to overcome the problems of scheduling alone. An integrated planning and scheduling system would be able to consider *alternative* sets of actions, unlike the stand–alone scheduler, which is unable to deviate from its given action set. We are working towards such an integrated system by incrementally constructing a unified theory of planning and scheduling that can be computationally expressed as practical software tools.

Our second subgoal is to study plan execution as a problem in *control* (Drummond, & Bresina 1990b). Most planning and scheduling work assumes that the job of the automatic system is done when a plan or schedule has been generated. Of course, one of the first things that you learn about plans is that they are rarely ever perfectly predictive of what will happen. As Dwight D. Eisenhower observed, "Plans are nothing, planning is everything". We agree with this view, since it tells us that the importance of planning does not lie in the existence of a *single* plan, but rather in a system's ability to re–plan and predictively manage plan execution failures in light of feedback from the environment. In the ERE project, we view plan execution as a problem in discrete event control; specifically, we formalize a plan as a simple type of feedback controller, and this gives us a

new view on plan execution. Traditionally, plans have been executed by executing each component action in sequence. In contrast, our plans are functions that map from current sensor values and a desired goal into a set of acceptable control actions. The interpretation of the function is that any of the actions, if executed in the current state, constitute an acceptable prefix to a sequence of actions that will eventually satisfy the goal.

4 The Current System

The previous two sections have, in rough terms, explained the APT problem and overall ERE project goals. In this section, we describe the current status of our efforts to use ERE technologies to produce an improved system for APT scheduling and control.

Recall that the job of a Principal Astronomer (PA) with respect to an Automatic Photoelectric Telescope (APT) is to collect together the observation requests of a number of telescope users and package them together into a single ATIS file for transmission to the telescope. The input to the PA can range from a verbal specification to actual ATIS request files. The PA has the job of looking over the set of observation requests from the individual users and attempting to predict or evaluate how these requests may cause the telescope to operate. For example, the PA might feel that there are too many high priority groups from a single user. In order to balance observation time for each user, the PA could modify priority levels of some groups in the hope that a fairer schedule will result.

An early phase of our project involved the development of tools to assist the PA with his or her current tasks. These decision support tools provide basic data management capabilities for browsing and editing a summarized form of the raw ATIS data. These initial tools have been modeled after a widely used PC–based program[2] designed to facilitate definition of observation requests (groups) and to provide various forms of data analysis on the ATIS result files returned from an APT. The challenging task from a PA's perspective is attempting to predict what the telescope will do with a given ATIS file. Current practice usually requires that the ATIS file, containing all the users' groups, be sent off to the telescope and used to control the telescope for several days or weeks. The actual execution behavior of the telescope is then evaluated by the PA, and if necessary, the ATIS input file is changed and sent back to the telescope. Once reasonably satisfactory telescope behavior is obtained, the PA is quite cautious about making radical changes to the input ATIS file.

The first step at improving current practice has involved the development of a predictive model of what groups the telescope will choose to execute. Since the group selection rules are well defined, it is relatively straightforward to duplicate them and predict, in advance, which groups will be selected for execution at the telescope. In essence, the predictive model can be used as a simulator for telescope operation.[3] Given an ATIS input file (and an initial assumption of clear weather), our "simulator" generates the sequence of groups (typically 50 to 100) which the telescope would observe throughout the evening. With this predictive tool, the PA now has the option to simulate execution of the ATIS file and then evaluate the resulting predicted time–line (using various 2D display plots). By incrementally modifying the ATIS input file, the PA can attempt to achieve a particular desired telescope behavior.

While the telescope simulation capability discussed above is quite useful, a user quickly realizes

[2]Designed and developed by George McCook of Villanova University.

[3]A related higher fidelity simulator for APTs has also been developed by AutoScope Corp.

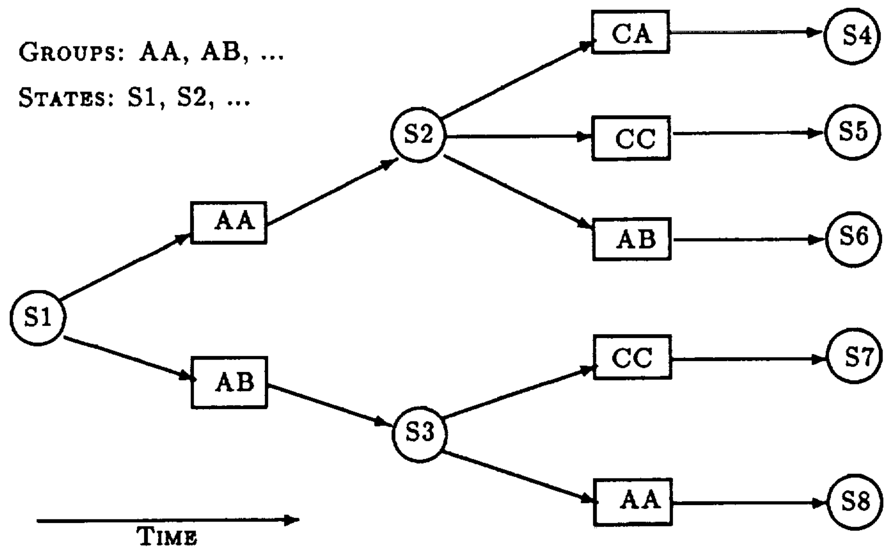

Figure 1: Forward Chronological Search Space for Schedules

that there are a *very* large number of possible telescope behaviors. Manually adjusting the ATIS file and simulating telescope operations is laborious, error-prone, and deeply uninteresting for most humans. Our approach is to allow the PA to mathematically define what he or she considers important in the eventual telescope behavior. This statement is sometimes called a "figure of merit"; in the scheduling literature, however, the statement is referred to as an "objective function". The objective function maps a sequence of groups (a possible schedule for a night) to a numeric score. The higher the score, the better the schedule. A realistic objective function is likely to be a weighted sum of various factors, possibly including the following attributes.

- How many high priority observations were selected?
- Did the users get roughly equal time?
- How close to the meridian were the observations?

This list is not intended to be exhaustive, and we are currently working with astronomers to better understand the various factors that should be included in a realistic objective function.

Assuming that we can evaluate a given schedule, how should possible schedules be generated? A wide range of techniques exist for generating schedules, and we do not have space here to discuss all the myriad ways (but see, for instance, Johnston, 1989; Muscettola, *et al.*, 1989; Collinot, *et al.*, 1988). Our approach to generating schedules is based on a "forward chronological search space". A small sample portion of the search space is shown in Figure 1. The search space is shown as a tree structure where a circular node represents the state of the telescope, and a rectangular node represents execution of a group by the telescope. The tree begins at a point in time early in the evening (*i.e.*, the state node labeled S1) and each possible schedule branches out into the future,

away from this unique starting point. For example, suppose that the duration of group AA was 5 minutes, the duration of group AB was 10 minutes, and that *either* group AA or group AB could be executed in state S1 (*i.e.*, both groups are enabled). Then the execution of these two different groups would lead to the two different successor states S2 and S3, as seen in Figure 1. Thus a transition from one state to another in this structure denotes the execution of a particular group, and the alternative branches out of any given state are all the groups enabled in that state. A *single* schedule is represented by a sequence of groups contained in a path from the root state of the tree (S1) to a leaf state (a state with no successors). For example, AA followed by CC would be a very simple schedule. Of course, the size of this branching structure is exponential in the number of groups, so it is impractical to exhaustively search it. This is no surprise, and one of our major short term goals is to apply what we already know about search control to this particular problem. Clearly, one possible approach is to use the value of a schedule under the given objective function to define an enumeration order for schedules in the tree.

Once a "preferred" schedule has been found, how is it communicated to the telescope? Also, how should the preferred schedule interact with the telescope controller's existing group selection rules? Our answer to this question is based on what we call the "principle of independent competence" (Bresina & Drummond, 1990). Briefly, in this context, the principle of independent competence requires that our scheduling system not degrade the baseline performance of the telescope controller. Thus, if the scheduler has a schedule that can improve the controller's operation, it should be used. However, in the absence of a better schedule, the default group selection rules should be used. This approach allows us to guarantee that our scheduler will never make the overall telescope performance worse than that obtained by the heuristic dispatch technique.

Our scheduler communicates schedules to the telescope controller in the form of Situated Control Rules, or SCRs (Drummond, 1989). An SCR is, basically, a rule: $S \rightarrow A$, with the interpretation "in state S take action A". For telescope control, the state corresponds to the time of night and a summary of group execution history; the action corresponds to the group that should be executed next. With this approach, a schedule can be represented as a set of SCRs and transmitted to the telescope along with the ATIS file. Of course, the local telescope control program must also be modified to accept the SCRs. In accordance with our principle of independent competence, the local telescope control program is modified so it uses an SCR if there is one available for the current state. Only if there is no currently applicable SCR will the existing group selection rules be used to select the next group. Thus, when the schedule can be followed, it is, but when the schedule "breaks", the telescope still operates using heuristic dispatch.

Our current implementation is capable of accepting ATIS files as input and can generate a wide range of alternative schedules in addition to the single "schedule" which would be produced by the group selection rules of the telescope controller. Several preliminary objective functions have been defined to evaluate schedule quality. Under any particular objective function, once the best schedule is found (in the time available), our system automatically constructs SCRs and transmits them to the telescope. At the moment, however, these SCRs have only been transmitted to an APT simulator developed by AutoScope Corporation. We have modified their simulator to accept and use SCRs in the manner described above.

5 Future Work

Our initial approach to the APT scheduling problem has been to produce a system that addresses the total range of required tasks. As a result, the sophistication of any individual component has been intentionally limited. Throughout this process, a number of interesting research issues have arisen, and we plan to study these as we increase the capability of each individual system component.

One very interesting problem is the robust execution of highly-tuned schedules. In scheduling, there is a recurrent tension between finding schedules of high quality (under some objective function) and schedules that are "robust". The robustness of a schedule relates to the ability of that schedule to withstand environmental perturbations: schedules that are of high quality tend to be rather brittle. For instance, if the objective function seeks to maximize the number of observations in a given night, then the resulting schedules will be tightly packed with many observations. If only one of these observations takes longer than expected the entire schedule can be in jeopardy. In contrast, the current approach taken by the ATIScope system, that of heuristic dispatch, is extremely robust with respect to environmental perturbations. The dispatch approach forms no expectations about the future, so it can hardly be disappointed when any given observation takes longer than it might otherwise. Indeed, the entire notion of "failure" is defined with respect to a specific prediction, so the heuristic dispatch approach can never fail, at least in this technical sense.

We have some preliminary ideas about how to manage the trade–off between schedule quality and robustness. One particular technique we are developing is called *Just–In–Case*, or JIC, scheduling (motivated by the term "Just–In–Time", or JIT). The basic idea involves explicitly considering the external events that could happen and, if they were to happen, how they could affect the predicted schedule. For each of the highest probability events, a number of contingent, or backup, schedules can be produced. Thus, instead of transmitting a single schedule to the telescope, a set of schedules is transmitted. Our SCR formalism handles this well, as it is easy to encode a set of contingent schedules as a set of state–action rules. We have an algorithm that can automatically generate such sets of SCRs (Drummond & Bresina, 1990a), but we do not yet know what modifications will be necessary for this algorithm to work on the telescope scheduling problem.

Finally, a longer range technical goal is to extend the specification language available to astronomers. Instead of having to be painfully specific about how to make observations, we feel that the astronomer should have the option of specifying observation goals and then let the scheduling system fill in the details. An advantage of this approach is that the automated system might be able to keep requesting specific observations until a higher level observation goal has been achieved. A possible first test case that we are considering involves a facility for filling out a hypothesized light curve. The automated system would continue to request observations until a specified sampling density over a star's period was achieved. Other test cases will be established in conjunction with our APT experts. The extra functionality offered at this stage of development will be that of *planning*, as opposed to pure *scheduling*. It is at this point that our system will really begin to offer increased scientific power over that of the traditional ATIScope–style system. Previous sections have only discussed how to increase the "quality" of the group execution sequences; here, we seek to increase the expressiveness of the language that is used by an astronomer to specify scientific objectives.

Of crucial importance to our efforts is getting actual operational telescope experience instead of just simulator time. To this end, we are purchasing, and intend to operate, a 16–inch APT.

This telescope will be located in northern California or Arizona, and will be made available to members of the scientific community, with the focus being on educational institutions. We will make our system available over the InterNet, such that remotely located astronomers can simply electronically mail request files to our system. The system will accept requests from various users, schedule them, and download the set of groups and SCRs to our telescope. Users will receive their requested data via return electronic mail or will be given access to an FTP site where their data may be retrieved. This will provide the first example of a completely automated telescope planning, scheduling, and control system. We hope to have a version of the system operating within 6 to 9 months.

Once individual APTs are routinely used by remotely located astronomers, with scheduling tasks performed automatically, many new opportunities arise. For instance, at this point it becomes practical to consider an electronic network of telescopes located around the world. One goal for such a network is the continuous observation of astronomical objects. While possible now for exceptional events (such as a supernova), the logistical overhead precludes wider practice. Our goal for the medium term is to demonstrate our system on such a network.

We hope that our demonstration of fully automatic telescope operations will serve to lay the groundwork for other sorts of astronomy. Of particular interest is the possibility of placing a number of small telescopes on the moon (Genet *et al.*, 1991). A lunar telescope facility would be an excellent test of our approach to simplified management structure. We feel that ERE can provide a solid basis for the development of integrated telescope planning, scheduling, and control systems that help to make this simplified management structure a reality.

Acknowledgements

Thanks to Bill Borucki, Russ Genet, David Genet, and Butler Hine for their help and enthusiasm.

References

[1] Bresina, J., Drummond, M., and Kedar, S. *Forthcoming.* Reactive, Integrated Systems Pose New Problems for Machine Learning. To appear in the volume on *Learning in AI Planning and Scheduling Systems*; Langley, P., and Minton, S. (eds).

[2] Bresina, J., and Drummond, M. 1990. Integrating Planning and Reaction: A Preliminary Report. *Proceedings of the AAAI Spring Symposium Series* (session on Planning in Uncertain, Unpredictable, or Changing Environments).

[3] Collinot, A., Le Pape, C., and Pinoteau, G. 1988. SONIA: a knowledge-based scheduling system. *Artificial Intelligence in Engineering*, Vol. 3., No. 2. pp. 86–94.

[4] Drummond, M. 1989. Situated Control Rules. *Proceedings of Conference on Principles of Knowledge Representation & Reasoning.* Toronto, Canada.

[5] Drummond, M., and Bresina, J. 1990a. Anytime Synthetic Projection: Maximizing the Probability of Goal Satisfaction. In proc. of *AAAI-90.*

[6] Drummond, M., and Bresina, J. 1990b. Planning for Control. In proc. of *Fifth IEEE International Symposium on Intelligent Control*, published by the IEEE Computer Society Press, Philadelphia, PA. pp 657–662.

[7] Drummond, M., Bresina, J., and Kedar, S. 1991. The Entropy Reduction Engine: Integrating Planning, Scheduling, and Control. *Proceedings of the AAAI Spring Symposium Series* (session on Integrated Intelligent Architectures).

[8] Genet, R.M, Genet, D.R., Talent, D.L., Drummond, M., Hine, B., Boyd, L.J., and Trueblood, M. 1991. Multi–Use Lunar Telescopes. A chapter in *Robotic Observatories in the 1990's*. Edited by Alexei V. Filippenko, published by the Astronomical Society of the Pacific Conference Series.

[9] Genet, R.M., and Hayes, D.S. 1989. *Robotic Observatories: A Handbook of Remote–Access Personal–Computer Astronomy*. Published by the AutoScope Corporation, Mesa, AZ.

[10] Hall, D.S., and Genet, R.M. 1988. *Photoelectric Photometry of Variable Stars*. Wilmann–Bell, PO Box 35025, Richmond, VA (2nd edition).

[11] Johnston, M. 1989. Reasoning with Scheduling Constraints and Preferences. Space Telescope Science Institute, SPIKE Technical Report No. 1989-2.

[12] Muscettola, N., Smith, S., Amiri, G., and Pathak, D. 1989. Generating Space Telescope Observation Schedules. Carnegie-Mellon University, Robotics Institute, CMU-RI-TR-89-28.